International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(04), December 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354

Tweaking recommendations using soft computing

Yash Shah[1], Manan Doshi[2], Neel Patel[3], Neil Patel[4]

[1,2,3,4]*Computer Engineering, Shri Bhagubhai Mafatlal Polytechnic (India)*

ABSTRACT

The world has begun to benefit from analysing large volumes of data. The enormous amount of data requires an efficient system to extract only relevant data based on user's preferences, opinions and perceptions. To attract more and more customers, companies have started to provide the facility of recommender systems (RSs) and personalized search engines that help users discover pertinent content from a diversified lot. This paper portrays content and collaborative filtering - the principles behind recommender systems. We also illustrate our ideology to implement personalized information sanctioning systems that use fuzzy logic to improve the efficiency of RSs.

Keywords: Recommender systems, fuzzy logic, user and item profiling, decision making

I. INTRODUCTION

We are marching towards an era of abundance from an era of scarcity. The web today is flooded with a plethora of information. New and innovative products are being introduced at a remarkable pace. This led to an information overload problem, thus making it difficult for users to choose from a massive variety of products. This resulted in poor decision making. Recommendation Systems (RSs) were established as a perfect answer to this drawback. These systems have proven to be efficient at retrieving relevant results from a substantial load of information. The reason behind their growing popularity lies in the fact that they extract data according to user’s taste.

Since their inception, the use of RS has extended quickly with existing applications that recommend movies, web-pages, news articles, medical treatments, music, and other products [2]. E-commerce giants such as amazon and eBay have employed RSs into their shopping experience. Recommender systems used in e-commerce are targeted marketing methods, which rely on historical experiences to increase the sales of products. [2]

RSs are built generally based on two different types of methods that are Content Based Filtering (CBF) and Collaborative Filtering (CF). The CBF approach creates content recommendations based on the characteristics of users or items. This system does not need data of other users and is able to recommend an item to users with unique taste. CF systems recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users. [4]

We suggest a conceptual framework for recommending one and only items based on fuzzy logic. The proposed fuzzy logic uses linguistic approach to capture the uncertainty in user preferences. It enables us to exploit the vague information in the domain.

The outline of the paper is organize as follows: Part II describes the fabrication of recommender systems. Part III portrays our proposed fuzzy approach at making recommendation.

II. BUILDING RECOMMENDER SYSTEMS

1. User information collection

To be able to relate to the user, RSs need to learn the user preferences. This is realized by collecting user information through implicit and explicit feedback. Explicit feedback relies on users to supply information about themselves through registration forms and questionnaires. They reflect the actual needs of the users. However they are an additional burden on the user and may raise privacy concerns. Thus encouraging users to provide information explicitly is a challenging task because without it the accuracy of the user profiles would be compromised.

On the other hand implicit feedback absorbs user behaviour through web usage logs, click stream, browsing histories, purchase records and content or structural information from visited web pages. This process does not require user interaction with the system. Then again converting behaviours into preferences is of major concern. [1]

2. Utility matrix

The collected data itself can be represented as a utility matrix, giving for each user-item pair, a value that represents what is known about the degree of preference of that user for that item. We assume that the matrix is sparse, meaning that most entries are "unknown." An unknown rating implies that we have no explicit information about the user's preference for the item.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	4			5	1		
B	5	5	4				
C				2	4	5	
D		3					3

Figure II-B: A utility matrix representing ratings of movies on a 1–5 scale [6]

3. Content based recommenders

Content based filtering approach utilizes contents of items as domain knowledge to develop user profiles. New items similar to user profile are recommended. User profiles play an important role in RSs as they mirror the user's preferences, needs and implicit and explicit interests. CBF determines the affinity between user and item by generating content related information on items that user rated, clicked, browsed or bought.

For each item we create an item profile which serves as a collection of features characterizing the item. Example: Movies: actor, title, director… People: set of friends

It is based on the item profiles that we build the user profile. The utility matrix serves as a reference for creating both the profiles.

International Journal of Advance Research in Science and Engineering
Volume No.06, Special Issue No.(04), December 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354

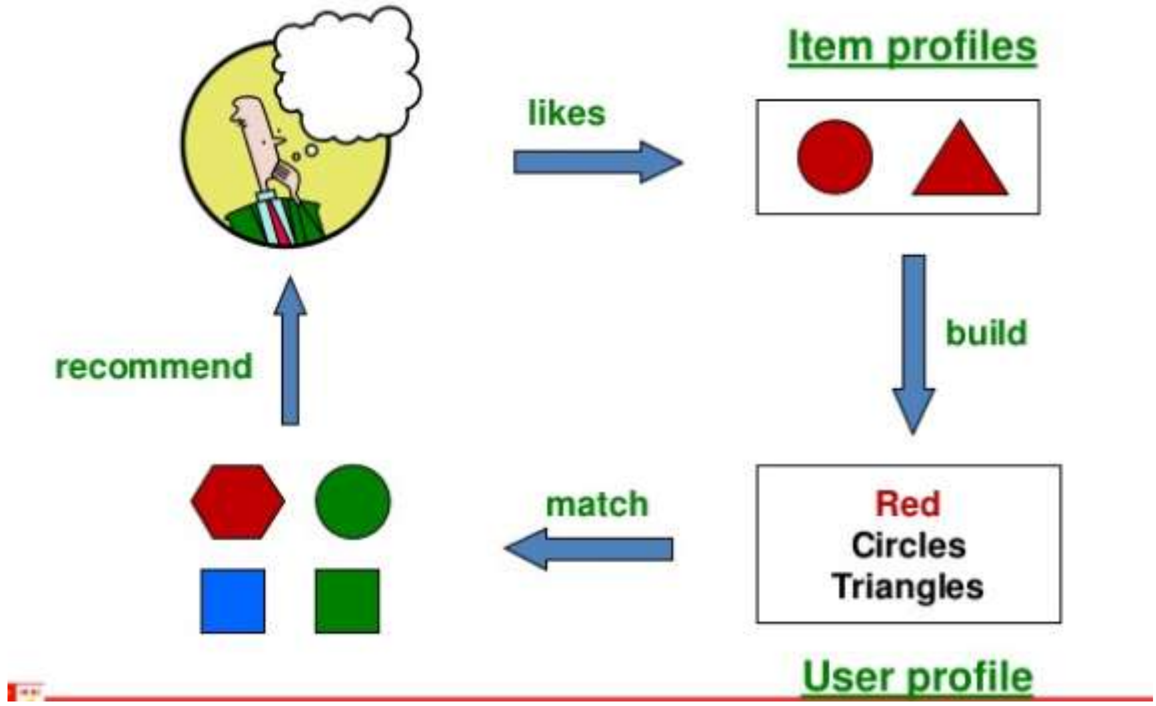

Figure II-C: CBF representation [1]

Our ultimate goal in content based filtering is to create both an item profile consisting of feature value pairs and user profiles summarizing user's preferences. We now illustrate each phase in a little more detail:

3.1. Representing item profiles Item

Item profiles can be thought of as vectors with one entry per feature. Each entry indicates feature's participation in the item. Consider an item Movie with only two features that is set of actors and the average rating. The vectors would appear like this:

$$0\ 1\ 1\ 0\ 1\ 1\ 0\ 1\ 3\alpha$$

$$1\ 1\ 0\ 1\ 0\ 1\ 1\ 0\ 4\alpha$$

The features of actors take only Boolean values indicating his/her participation in the movie. The two movies have two actors in common and five actors each. 3α and 4α indicate user rating. Actors not featuring in any of the movie do not pose a problem as they do not affect the cosine similarity as discussed later.

3.2. Building item profile for text features

In recommending news articles for instance the values of features may not be immediately known. In this case the profile consists of important words in the document. For this purpose we make use of a concept called TF-IDF that is Term Frequency v/s Inverse Document Frequency. It is a statistical measure of determining a word's value.

TFij = fij/maxkfkj

TF indicates frequency of occurrence of a word in the document.

IDF i determines the word's significance. n i is the number of documents containing the word whereas N is the total number of documents. Thus a composite weight w ij is assigned to each word of meaning to us:

w ij = TF ij x IDF i

We can now create a doc profile with a set of words having high TF-IDF scores with their corresponding weights.

3.3. Building user profile

The accuracy of a user profile plays a great role in the performance of RSs. User profiling can be achieved in two ways- knowledge based and behaviour based approach. While the former emphasizes on implicit and explicit domain knowledge, the latter utilizes user behaviour to exploit useful patterns using machine learning.

Here we make use of Vector Space Model VSM for representation where both user profiles and item profiles are characterized as vectors.

Consider that the utility matrix for the item movie is a multivalued vector which takes ratings from 1-5 then we can weigh the vectors representing the profiles of items by the utility value. For example consider the following ratings by a user:

A's movies were rated 3 and 5

B's movies were rated 1, 2 and 4

Assuming the average rating of the user to be 3, we normalize each user rating with this average value.

Normalized ratings of A: (3 - 3), (5 - 3) 0 +2

Profile weight: (0 + 2) / 2 = 1

Normalized ratings of B: (1 - 3), (2 - 3), (4 - 3) -2 -1 1

Profile weight: (-2 -1 + 1) / 3 = -2/3

So we obtain negative weights for items with below average rating and positive weights for above average rating. Our example indicates that the user likes actor A's movies more than B.

3.4. Making recommendations

With profile vectors for both users and items we can estimate the degree to which a user would prefer an item by computing the cosine similarity between users and item vectors. Given user profile X, item profile I:

Estimate U (X, I) = cosine (Θ) = (X. I) / (|X|.|I|) Smaller the angle Θ larger is cosine (Θ) and more likely is the user to rate the item highly. All the item profiles are placed in buckets. We search the buckets for items that have a small cosine distance from the user. [1]

4. Collaborative filtering

Collaborative filtering is a significantly different approach at making recommendations. It filters information using the recommendations of other people. It is based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future. Instead of contriving a profile vector for users, we represent them by their rows in the utility matrix.

IJARSE
ISSN (O) 2319 - 8354

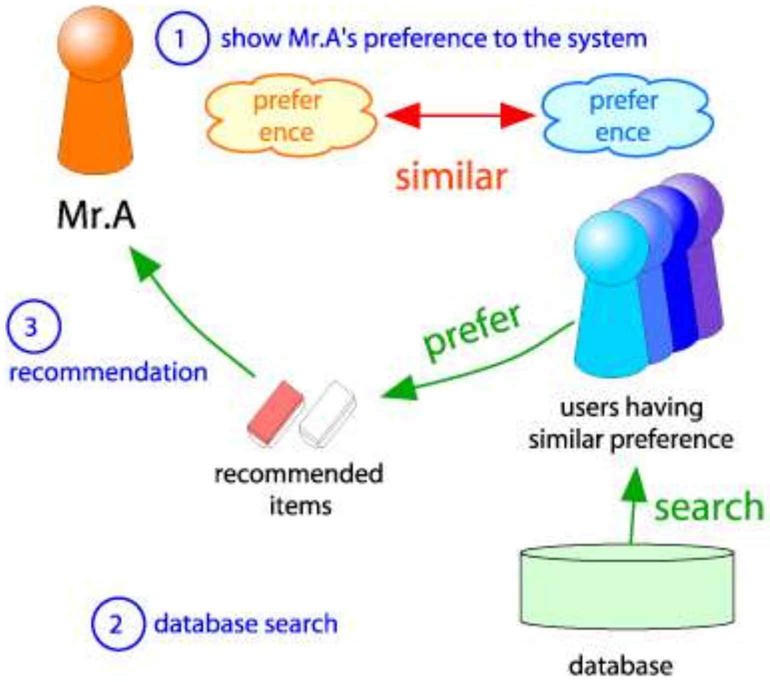

Figure II-D: CF representation [1]

Recommendation for a user is made by searching for users that have similar taste and recommending items that these users like. We now demonstrate each phase in a little more detail.

4.1. Measuring similarity

The next step is to measure the resemblance among the users. The figure given below is a matrix that represents users on the y-axis and their ratings for the movies on x-axis. We can relate that users A and B have similar perception but users A and C have different one. If there are two users ‘x’ and ‘y’, we need a similarity metric Sim(x, y) that gives us the measure of their similarity. We need to capture the intuition that Sim (A, B) >Sim (A, C). In other words the similarity metric of similar users should be greater than that of dissimilar ones. The methods Jaccard distance and cosine similarity failed to achieve best approximation. Centred cosine proved to be efficient than the former two.

4.2. Centred cosine

An efficient way to measure the similarity is to normalize the ratings by subtracting the row mean from each rating. The unknown ratings are assumed to be as zero. The resultant matrix after modification is shown in figure 2.6.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
A	$2/3$			$5/3$	$-7/3$		
B	$1/3$	$1/3$	$-2/3$				
C				$-5/3$	$1/3$	$4/3$	
D		0					0

Figure II-D 2.1: modified utility matrix -2 [6]

IJARSE
ISSN (O) 2319 - 8354

The point to be noted is that we have centred each rating on zero thus instigating the mean of each row to be zero. Positive rating indicates that the user liked the movie more than average and negative rating indicates that the user likes the movie less than average. The magnitude of rating also indicates how much the user liked or disliked the movie. Now we can compute cosine similarity using these centred ratings.

Similarity between users A and B is given as:

(Sim (A,B)= cos(rA,rB) :

$$\frac{(2/3)\times(1/3)}{\sqrt{(2/3)^2+(5/3)^2+(-7/3)^2}\sqrt{(1/3)^2+(1/3)^2+(-2/3)^2}} = 0.092$$

Figure II-D 2.2

In a similar manner the cosine of angle between A and C can be given as cos (rA, rC) :

$$\frac{(5/3)\times(-5/3)+(-7/3)\times(1/3)}{\sqrt{(2/3)^2+(5/3)^2+(-7/3)^2}\sqrt{(-5/3)^2+(1/3)^2+(4/3)^2}} = -0.559$$

Figure II-D 2.3

This actually captures the fact that A and C are dissimilar users. Also notice that there is a big gap between the cosines of A & B and A & C which reveals that A and B are more similar users than A and C. The centred cosine captures the intuition better than the simple cosine. This is because in the former case the missing ratings were treated as negative rating and the latter treats the missing ratings as average ratings. It effectively handles tough raters and easy raters.

4.3. Making recommendations

Let rx be the vector of user x's rating, we are going to use the notion of centred cosine similarity to find a set 'N' of users called as neighbourhood. The neighbourhood consists of users who are more similar to user x who have also rated item 'i'. We go through all the users to find the similarity between it and the user 'x' and select k users with the highest similarity values to constitute set N.

The first and simple way is to find out the average of ratings in the set n and predict it as the rating of user 'x'. Despite being simple it ignores the similarity values between the users.

The second option is to find out the weighted average. For each user 'y' in the neighbourhood of 'x' we weight y's rating for item 'i' by the similarity of user 'x' and 'y' and normalize it by taking the sum of similarities.[1][2]

III. PROPOSED FUZZY APPROACH

Recommender systems are in a continuous need for personalization in order to make effective suggestions and to provide valuable information of items available. Model-based collaborative methods and content-based based methods present some problems. Consequently current approaches are likely to employ concepts from both categories of methods in order to take benefit from the strengths of each of them.

IJARSE
ISSN (O) 2319 - 8354

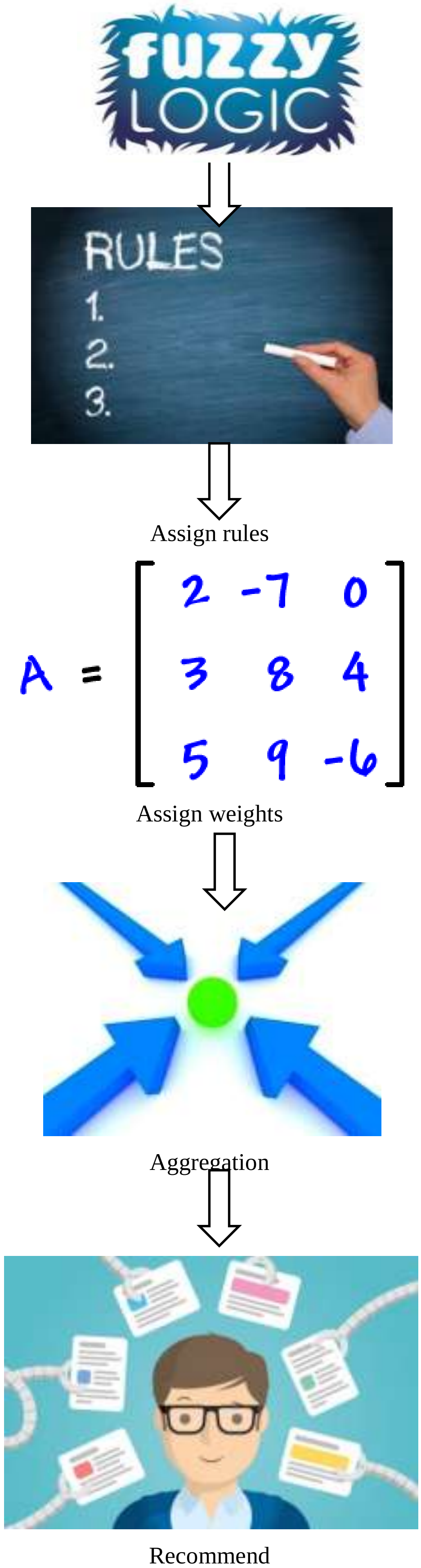

Fig 3: Proposed model

IJARSE
ISSN (O) 2319 - 8354

Our goal is to design an efficient CF algorithm that uses preferences of similar learners (neighbours) to predict the active learner's preferences, then, generating diversified recommendations that meet their needs according to his preferences membership degrees to improve their top-N recommendation performance. These membership values can be obtained in the CF phase by what we present is the type-2 fuzzy logic.

Type-2 fuzzy logic alleviates the preferences' vagueness problem of RSs and models the variations in human decision making. This facilitates improved quality and effectiveness.

A. Build Multi criteria User Item Matrix

Multi criteria ratings offer more information about user's preferences from different aspects of an item and lead to more accurate recommendation than a single-rating system. Items are then recommended to the users based on the multi criteria ratings provided by others users. For example, the single user rating for music gives the general user preference on that music. However, the multi criteria ratings of a music such as ratings for music, lyric and voice provide in-depth knowledge about the user preferences on that music.

One of the important steps in fuzzy recommender system is defining weights. In formal, multi criteria user-item ratings matrix is a matrix of size M x N, where the element Ritj (i=1, 2,.., m; t =1, 2,…, k ; j=1, 2,…, n) is the rating assigned to alternative item Ii by the user U j under criterion Ct . Wtj is the weight given to criteria Ct by user U j, m is the total number of item, n being the total number of users, and k is the total number of criteria.

$$
\begin{array}{cc}
 & \begin{array}{cccccc} & C_1 & C_2 & C_3 & \cdots & C_t \\ Items & W_1 & W_2 & W_3 & \cdots & W_t \\ \hline \end{array} \\
U_1 & \begin{pmatrix} I_1 & R_{11} & R_{12} & R_{13} & \cdots & R_{1t} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ I_m & R_{m1} & R_{m2} & R_{m3} & \cdots & R_{mt} \end{pmatrix}
\end{array}
$$

Figure III A: Multi criteria user item matrix [2]

The scale normally ranges from 1 to 5, where 1 denotes the greatest dislike to the item and 5 denotes the greatest like to the item. Here, the linguistic assessment is used instead of numerical value representation. Instead of specifying numerical scale while collecting feedback, the linguistic terms are used to collect. Due to the subjective, imprecise and vague nature of user preference data, the fuzzy linguistic approach is adopted to represent the user's preferences.

B. B. Fuzzifying Multi Criteria User-Item Matrix

After obtaining user ratings on items in multiple aspects (criteria) and weights for different criteria in the form of preference rating matrix; these values are fuzzified to determine Triangular Fuzzy Number (TFN) - degree of membership in the user preference fuzzy set. The users find it easy to express their preferences on item for different criteria using natural language terms rather than numerical values.

IJARSE
ISSN (O) 2319 - 8354

Fuzzy sets serve to be the most suitable tool for handling dynamic behaviour. Here, the customer supplies his tastes, preferences and opinions in qualitative form; the system turns it into their respective quantitative forms using the concept of type-2 fuzzy logic.

A type-2 (T2) FS is characterized by a fuzzy membership function, i.e., the membership value (or membership grade), for each element of this set is a FS in [0, 1], unlike a type-1 FS where the membership grade is a crisp number in [0, 1]. The membership functions of T2 FSs are three dimensional and include a footprint of uncertainty (FOU), that makes it possible to directly model and handle uncertainties.

No.	Linguistic Term	TFN
1	Very unlikely (VUL)	(0, 1, 2)
2	Unlikely (UL)	(1, 2, 3)
3	Medium (M)	(2, 3, 4)
4	Likely (L)	(3, 4, 5)
5	Very likely (VL)	(4, 5, 6)

Figure III B2: 3 Linguistic set and TFN [2]

In the new concepts, there are upper membership function and lower membership function that represented by T1FS membership function. The area between these two functions is footprint of uncertainty, which is used to characterize T2FS.

$$LMF = \begin{cases} h(x+a)/a, & \text{if } -a \le x \le 0 \\ h(a-x)/a, & \text{if } \quad 0 \le x \le a \\ 0, & \text{otherwise} \end{cases}$$

$$UMF = \begin{cases} h(x+b)/b, & \text{if } -b \le x \le 0 \\ h(b-x)/b, & \text{if } \quad 0 \le x \le b \\ 0, & \text{otherwise} \end{cases}$$

Figure III B1: LMF & UMF [2]

Here 0 <=a <=b and 0 <=h <=1. All of MF' parameters are numerically specified based on the experiences. In our case five linguistic sets presented in Table 1 are used to enable users express their opinion for each criteria and weight: Very unlikely (VUL), Unlikely (UL), Medium (M), Likely (L), and Very likely (VL).

The aggregation of different ratings is done by simply multiplying the ratings and the fuzzy weights. The distance between each alternative item and the positive ideal solution as well as negative ideal solution is calculated. The chosen alternative should have the shortest geometric distance from the positive ideal solution and the longest geometric distance from the negative ideal solution. After obtaining the ranking value of each item, sort the values of in descending sequence. The list of recommendations to be generated is chosen by selecting the Top-N items with the highest scores.

Fuzzy RSs take into account user preferences based on multiple criteria thereby improving the accuracy of the recommendations. The major drawback of the presented fuzzy system is that the cold start problem remains unresolved. Also the user here is burdened with supplying explicit information. This in a way may hamper privacy. Users may not be willing to share so much information. This might degrade the performance of RSs.

VI. CONCLUSION

Soft computing appears to be a suitable model to handle the uncertainty and fuzziness on user preference and to efficiently model the natural complexity of human behaviour. To improve the recommendation quality, we are conducted towards fusion of CF and type-2 fuzzy linguistic modelling. This paper adopts high order fuzzy linguistic approach to represent the user preferences and multi criteria decision making method to rank the appropriate, relevant items to a user in CF recommendation context. It can capture the hidden connections between users and items and have the ability to provide unexpected items, which are helpful to improve the diversity of recommendation. [3][4]

REFERENCES

[1.] Anon, (2017). Mining of massive datasets. [online] Available at: http://infolab.stanford.edu/~ullman/mmds/book.pdf [Accessed 9 Dec. 2017.

[2.] Darwish, Saad. (2014). Contribution to Collaborative Filtering Based on Soft Computing to Enhance Recommender System for e-Commerce. International Journal of e-Education, e-Business, e-Management and e-Learning. 4. . 10.7763/IJEEEE.2014.V4.341.

[3.] A fuzzy logic based personalized recommender system. (2017). International journal of Computer Science and Information Technology & Security 2(5), pp.1008-1015.

[4.] An exploration of improving collaborative recommender systems via user-item subgroups. (2017). Proceeding of The 21st International Conference on World Wide Web.

[5.] Slideshare.net. (2017). Recommender systems: Content-based and collaborative filtering. [online] Available at: https://www.slideshare.net/microlife/recommender-systems-contentbased-and-collaborative-filtering [Accessed 9 Dec. 2017].

[6.] Zenebe, A., & Norcio, A. F. (2009). Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems. Fuzzy Sets and Systems, 160(1), 76-94.

[7.] Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multi-criteria rating systems. IEEE Trans. Intelligent Systems, 22(3), 48-55.

[8.] Naim, S., & Hagras, H. (2012). A hybrid approach for multi-criteria group decision making based on interval type-2 fuzzy logic and intuitionistic fuzzy evaluation. Proceedings of the IEEE International Conference on Fuzzy Systems (pp. 1-8).

[9.] G. Adomavicius and A. Tuzhilin, "Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," IEEE Trans. on Data and Knowledge Engineering 17:6, pp. 734–749, 2005.

[10.] Jerry, M. (2007). Type-2 fuzzy sets and systems: an overview. IEEE Computational Intelligence Magazine,2(1), 20-29.

[11.] Liang, Q., & Mendel, J. M. (2000). Interval type-2 fuzzy logic systems: theory and design. IEEE Transactions on Fuzzy Systems, 8(5), 535–550.